UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/779,808	02/18/2004	Paul Anthony Gilkerson	550-525	6826

23117 7590 10/16/2006

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
FENNEMA, ROBERT E	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 10/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/779,808	GILKERSON, PAUL ANTHONY	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-21 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-6, 10-16, and 20-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Furber, in view of Patterson.

4. As per Claim 1, Furber teaches: A data processing apparatus, comprising:

a processor operable to execute a stream of instructions (Page 387, the AMULET 3);

a prefetch unit operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution (Page 387), the prefetch unit being operable to receive from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory (Page 387, it can fetch two Thumb instructions), and being operable to detect whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address for a next instruction to be prefetched by the prefetch unit (Page 382, where it

says it modifies the predicted control flow to the predicted target address. Page 387 says that the prefetch unit in the AMULET3 functions similar to the jump trace buffer shown on 382, but to support the multiple instructions, and that a "hit" indicates that it is a branch instruction); and

address generation logic within the prefetch unit and responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction, for determining a target address to be output as the fetch address (Pages 382-383), the address generation logic having a first address generation path for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality (Page 388), and at least one further address generation path for determining the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality, the first prefetched instruction being earlier in said stream than said other prefetched instructions (Page 388), the first address generation path generating the target address more quickly than the at least one other further address generation path (Page 388, the first path takes priority); and

whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit outputs the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction (Page 388, the first path takes priority), but fails to teach:

a pipeline stage, provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path.

However, Patterson teaches that pipelining a processor, or adding stages of a pipeline, will reduce CPI, and reduce clock cycle time (increasing the rate at which the clock runs) (Page A-3), which would then speed up execution of the first address path, due to the increased clock rate. Given this advantage, it would have been obvious to one of ordinary skill in the art at the time the invention was made to pipeline a processing path that took one long clock cycle, to increase the clock rate for other paths, and apply it to Furber's invention.

5. As per Claim 2, Furber teaches: A data processing apparatus as claimed in claim 1, further comprising:

prediction logic operable to predict, for a prefetched instruction whose execution is conditional, whether that conditional prefetched instruction will be executed by the processor (Pages 382-383);

in the event that the first prefetched instruction is said selected prefetched instruction and its execution is conditional, the prefetch unit being operable to output the associated target address as the fetch address if the prediction logic predicts that the first prefetched instruction will be executed by the processor (Pages 382-383).

6. As per Claim 3, Furber teaches: A data processing apparatus as claimed in claim 1, wherein the prefetch unit associates a different priority level with each of the plurality of prefetched instructions, the first prefetched instruction having the highest priority level, and if more than one of the plurality of prefetched instructions is detected to be

said instruction flow changing instruction, the prefetch unit is operable to determine as said selected prefetched instruction the prefetched instruction having the higher priority level from those prefetched instructions detected to be said instruction flow changing instruction, whereby the target address associated with that selected prefetched instruction is output as the fetch address (Page 388).

7. As per Claim 4, Furber teaches: A data processing apparatus as claimed in claim 1, wherein the address generation logic is operable to generate the target address for the first prefetched instruction in a same clock cycle as the prefetch unit detects that that first prefetched instruction is said instruction flow changing instruction (Page 382).

8. As per Claim 5, Furber teaches: A data processing apparatus as claimed in claim 1, wherein predetermined logic is shared between the first address generation path and the at least one further address generation path (Page 387, a memory).

9. As per Claim 6, Furber teaches: A data processing apparatus as claimed in claim 1, wherein the at least one further address generation path comprises a single further address generation path used to determine the target address for any prefetched instructions other than said first prefetched instruction (Pages 387-388).

10. As per Claim 10, Furber teaches: A data processing apparatus as claimed in claim 1, wherein if none of the plurality of prefetched instructions is said instruction flow

changing instruction, the prefetch unit is operable to generate the fetch address by incrementing a previous fetch address output by the prefetch unit (Page 382, figure 14.6).

11. As per Claim 11, Furber teaches: A method of operating a data processing apparatus to determine a target address for an instruction flow changing instruction, the data processing apparatus having a processor operable to execute a stream of instructions (Page 387, the AMULET 3), and a prefetch unit operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution (Page 387), and to output a fetch address for a next instruction to be prefetched from the memory (Page 382, where it says it modifies the predicted control flow to the predicted target address. Page 387 says that the prefetch unit in the AMULET3 functions similar to the jump trace buffer shown on 382, but to support the multiple instructions, and that a "hit" indicates that it is a branch instruction), the method comprising the steps of:

(a) receiving from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory (Page 387, it can fetch two Thumb instructions);

(b) detecting whether any of those prefetched instructions are an instruction flow changing instruction (Page 382, where it says it modifies the predicted control flow to the predicted target address. Page 387 says that the prefetch unit in the AMULET3

functions similar to the jump trace buffer shown on 382, but to support the multiple instructions); and

(c) for a selected prefetched instruction that is detected to be said instruction flow changing instruction, determining a target address to be output as the fetch address (Pages 382-383) by performing one of the steps of:

(c)(1) employing a first address generation path to determine the target address if the selected prefetched instruction is a first prefetched instruction in said plurality (Page 388); or

(c)(2) employing at least one further address generation path to determine the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality (Page 388); the first prefetched instruction being earlier in said stream than said other prefetched instructions, and the first address generation path being arranged to generate the target address more quickly than the at least one other further address generation path (Page 388, the first path takes priority);

(e) outputting as the fetch address the target address generated at step (c); whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit is operable to output the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction (Page 388, the first path takes priority), but fails to teach:

(d) providing a pipeline stage in the at least one further address generation path in order to increase speed of generation of the target address by the first address generation path.

However, Patterson teaches that pipelining a processor, or adding stages of a pipeline, will reduce CPI, and reduce clock cycle time (increasing the rate at which the clock runs) (Page A-3), which would then speed up execution of the first address path, due to the increased clock rate. Given this advantage, it would have been obvious to one of ordinary skill in the art at the time the invention was made to pipeline a processing path that took one long clock cycle, to increase the clock rate for other paths, and apply it to Furber's invention.

12. As per Claim 12, Furber teaches: A method as claimed in claim 11, further comprising the step of:

employing prediction logic to predict, for a prefetched instruction whose execution is conditional, whether that conditional prefetched instruction will be executed by the processor (Pages 382-383);

in the event that the first prefetched instruction is said selected prefetched instruction and its execution is conditional, the prefetch unit being operable to output the associated target address as the fetch address if the prediction logic predicts that the first prefetched instruction will be executed by the processor (Pages 382-383).

13. As per Claim 13, Furber teaches: A method as claimed in claim 11, further comprising the steps of:

associating a different priority level with each of the plurality of prefetched instructions, the first prefetched instruction having the highest priority level (Page 388);

if more than one of the plurality of prefetched instructions is detected to be said instruction flow changing instruction, determining as said selected prefetched instruction for said step (c) the prefetched instruction having the higher priority level from those prefetched instructions detected to be said instruction flow changing instruction (Page 388);

whereby at said step (d) the target address associated with that selected prefetched instruction is output as the fetch address (Page 388).

14. As per Claim 14, Furber teaches: A method as claimed in claim 11, wherein at said step (c)(1) the target address for the first prefetched instruction is generated in a same clock cycle that, during said step (b), that first prefetched instruction is detected as said instruction flow changing instruction (Page 382).

15. As per Claim 15, Furber teaches: A method as claimed in claim 11, wherein predetermined logic is shared between the first address generation path and the at least one further address generation path (Page 387, a memory).

16. As per Claim 16, Furber teaches: A method as claimed in claim 11, wherein the at least one further address generation path comprises a single further address generation path used at said step (c)(2) to determine the target address for any prefetched instructions other than said first prefetched instruction (Pages 387-388).

17. As per Claim 20, Furber teaches: A method as claimed in claim 11, wherein if none of the plurality of prefetched instructions is determined at said step (b) to be said instruction flow changing instruction, the method further comprises the step of:

generating the fetch address by incrementing a previous fetch address output by the prefetch unit, and outputting that fetch address at said step (d) (Page 382, figure 14.6).

18. As per Claim 21, Furber teaches: A prefetch unit for a data processing apparatus that has a processor operable to execute a stream of instructions (Page 387, the AMULET 3), the prefetch unit being operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution (Page 387), the prefetch unit being operable to receive from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory (Page 387, it can fetch two Thumb instructions), and being operable to detect whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address for a next instruction to be prefetched by the prefetch unit (Page 382, where it says it modifies the predicted control flow to the predicted target address. Page 387 says that the prefetch unit in the AMULET3 functions similar to the jump trace buffer shown on 382, but to support the multiple instructions, and that a "hit" indicates that it is a branch instruction), the prefetch unit comprising:

address generation logic, responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction, for determining a target

address to be output as the fetch address (Pages 382-383), the address generation logic having a first address generation path for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality (Page 388), and at least one further address generation path for determining the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality (Page 388), the first prefetched instruction being earlier in said stream than said other prefetched instructions, the first address generation path generating the target address more quickly than the at least one other further address generation path (Page 388, the first path takes priority); and

whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit is operable to output the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction (Page 388, the first path takes priority), but fails to teach:

a pipeline stage, provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path.

However, Patterson teaches that pipelining a processor, or adding stages of a pipeline, will reduce CPI, and reduce clock cycle time (increasing the rate at which the clock runs) (Page A-3), which would then speed up execution of the first address path, due to the increased clock rate. Given this advantage, it would have been obvious to one of ordinary skill in the art at the time the invention was made to pipeline a

processing path that took one long clock cycle, to increase the clock rate for other paths, and apply it to Furber's invention.

19. Claims 7-9 and 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Furber and Patterson, further in view of Hara (USPN 5,848,269).

20. As per Claim 7, Furber teaches: A data processing apparatus as claimed in claim 1, wherein the prefetch unit comprises decode logic operable to detect whether any of the plurality of prefetched instructions are said instruction flow changing instruction (Pages 382-383), but fails to explicitly teach:

the decode logic further being operable to decode from each prefetched instruction detected to be said instruction flow changing instruction an immediate value to be input to the address generation logic.

While Furber teaches a prefetch unit to detect whether there is an instruction flow changing instruction using a branch prediction unit, and outputting a target address, it is not explicitly taught that the branch prediction unit can handle branches with immediate values. However, Hara teaches a branch prediction mechanism, which can be used for effectively calculating branch targets, without adding excessive hardware (Column 5, Lines 18-24), capable of effectively predicting without a correlating past history (Column 5, Lines 1-5), which has a method for predicting branches with an immediate value, and outputting a target address (Column 8, Lines 44-63, Column 19, Line 58 – Column 20, Line 4, also see Figure 15). Given that branches with immediate targets exist and can

be encountered in the instruction stream, and given a need to handle them in Furber's invention, one of ordinary skill in the art at the time the invention was made would have recognized the advantage of using a branch prediction unit such as Hara's, to take advantage of handling immediate values in a branch, while minimizing extra hardware, and being able to effectively predict branches even without correlation with past results.

21. As per Claim 8, Hara teaches: A data processing apparatus as claimed in claim 7, wherein the address generation logic comprises adder logic operable to determine the target address for the selected prefetched instruction by adding the associated input immediate value to the address of that selected prefetched instruction (Figure 15, and Column 19, Line 58 – Column 20, Line 4).

22. As per Claim 9, Furber and Hara teach: A data processing apparatus as claimed in claim 8, wherein the adder logic is shared between the first address generation path and the at least one further address generation path (Hara, Column 19, Line 58 – Column 20, Line 4 discloses the adder, and Furber, Page 388 discloses that one path takes priority, meaning only one target can be output at a time, meaning the adder must be shared).

23. As per Claim 17, Furber teaches: A method as claimed in claim 11, wherein the prefetch unit comprises decode logic operable at said step (b) to detect whether any of

the plurality of prefetched instructions are said instruction flow changing instruction (Pages 382-383), but fails to explicitly teach: the method further comprising the step of:

employing the decode logic to decode from each prefetched instruction detected to be said instruction flow changing instruction an immediate value for use in said step (c).

While Furber teaches a prefetch unit to detect whether there is an instruction flow changing instruction using a branch prediction unit, and outputting a target address, it is not explicitly taught that the branch prediction unit can handle branches with immediate values. However, Hara teaches a branch prediction mechanism, which can be used for effectively calculating branch targets, without adding excessive hardware (Column 5, Lines 18-24), capable of effectively predicting without a correlating past history (Column 5, Lines 1-5), which has a method for predicting branches with an immediate value, and outputting a target address (Column 8, Lines 44-63, Column 19, Line 58 – Column 20, Line 4, also see Figure 15). Given that branches with immediate targets exist and can be encountered in the instruction stream, and given a need to handle them in Furber's invention, one of ordinary skill in the art at the time the invention was made would have recognized the advantage of using a branch prediction unit such as Hara's, to take advantage of handling immediate values in a branch, while minimizing extra hardware, and being able to effectively predict branches even without correlation with past results.

24. As per Claim 18, Hara teaches: A method as claimed in claim 17, wherein at said step (c) the target address for the selected prefetched instruction is determined by

adding the associated immediate value to the address of that selected prefetched instruction (Figure 15, and Column 19, Line 58 – Column 20, Line 4).

25. As per Claim 19, Furber and Hara teach: A method as claimed in claim 18, wherein adder logic used to perform said adding step is shared between the first address generation path and the at least one further address generation path (Hara, Column 19, Line 58 – Column 20, Line 4 discloses the adder, and Furber, Page 388 discloses that one path takes priority, meaning only one target can be output at a time, meaning the adder must be shared).

Response to Arguments

26. Applicant's arguments filed 7/31/2006 have been fully considered but they are not persuasive.

Firstly, Applicant has essentially argued that Furber fails to teach the limitation of a first and a further address generation path in the independent claims, and has asked Examiner to clarify where this is taught in Furber. Examiner directs Applicant to Page 388, in the first paragraph, where it is stated that a Thumb packet contains two instructions, and that should both of them be branches, one must take priority, as the branch prediction unit can only handle one instruction at a time. As two instructions are fetched, there are two address generation paths, which can further be seen in Figure 14.10, as there are multiple inputs into the Execute stage. Furthermore, given that two instructions can be fetched, but that only one branch can be interpreted at a time,

Examiner interprets as further evidence that there are multiple "paths", one for each instruction, however, only one path can be used for a branch (at a time). Applicant has further argued that there is no generation of an address "more quickly than the other", and Examiner disagrees. As one path takes priority over the other, and produces a target address preferentially from that path over the other, it clearly is produced sooner in time than the other, thus more quickly.

Addressing Claim 21, a typographical error in the 102 summary sentence omitted the rejection of the Claim under 102 in view of Furber, this has been corrected in this action, but it is believed that given it's placement with the other 102 rejections, and the detailed rejection using Furber as a reference was enough to avoid any confusion over how it was rejected, which Applicant appears to have recognized.

As per the 35 USC 103 rejections, Applicant has argued that Examiner has not provided any motivation to combine the references. These motivations were provided in the rejections of the claims under 35 USC 103, and Applicant can examine the claims in question to read said motivations.

Finally, Examiner acknowledges the explanation of Claims 5, 9, 15, and 19 in regards to the 35 USC 112 rejections, and has withdrawn said rejections in light of Applicants arguments. The Examiner also acknowledges the substitute drawings and abstract, and has withdrawn the objections to each.

Conclusion

27. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100